



Renzo DF (RenzoDF)

SCILAB E NUMERI COMPLESSI

13 November 2008

Abstract

Alcune considerazioni sull'algebra complessa che, partendo dalle funzioni di base disponibili in Scilab, portano ad uno script finale utilizzabile nei calcoli di reti in c.a.

Introduzione

In elettrotecnica, nella soluzione delle reti in regime periodico alternato sinusoidale (PAS), grazie al lavoro di Steinmetz [1] e Kennelly di fine ottocento, si fa ancor oggi uso estensivo del metodo simbolico.

Le "Hyperbolic Tables" di Kennelly [2] sono, per fortuna, sparite ma rimane la sua notazione **polare** per i numeri complessi [3].

$$\bar{Z} = \rho \angle \vartheta$$

La disponibilit  odierna di potenti strumenti di calcolo e di software dedicato ci permette di evitare un sacco di lavoro e questo articolo cerca di analizzare le possibilit  di calcolo usando il software [Scilab](#).

Scilab implementa il calcolo complesso con una serie di funzioni predefinite ma, per chi   abituato alla notazione polare, presenta subito un problema di adattamento; da buon software matematico, impone l'uso di una delle due scritture care a Steinmetz:

$$\bar{Z} = \rho e^{j\vartheta}$$

detta **esponenziale**, che in Scilab diventa

```
-->z=rho*exp(%i*theta)
```

oppure

$$\bar{Z} = \rho(\cos \theta + j \sin \theta)$$

detta **trigonometrica**, che in Scilab si scrive

```
-->z=rho*(cos(theta)+%i*sin(theta))
```

L'equivalenza fra le due rappresentazioni risiede come noto nella famosa formula di Eulero $e^{j\vartheta} = \cos \vartheta + j \sin \vartheta$

Si deve comunque ricordare che, mentre nella polare l'angolo viene normalmente espresso in gradi, nella esponenziale sono d'obbligo i radianti. Di conseguenza se vogliamo trasformare partendo da una fase "theta" in gradi si dovrà scrivere

```
-->z=rho*exp(%i*theta*pi/180)
```

una scrittura senza dubbio scomoda e inusuale per molti.

In elettrotecnica poi si impone spesso la necessità di trasformazione dalla forma:

Cartesiana a Polare

In Scilab esistono le funzioni **abs(z)**, **phasemag(z)** che restituiscono modulo e fase (in gradi) e **polar(z)** che può essere usata per ricavarli contemporaneamente (con fase in radianti)

```
--> [rho,theta]=polar(z)
```

Unico problema il noioso comparire, nella fase, di una componente immaginaria vicina allo zero macchina. La suddetta è però inutilizzabile se z è un vettore riga o colonna !

Usando invece **abs()** e **phasemag()** possiamo definire una funzione del tipo:

```
//----- cartesiana --> polare -----
function[mf]=cp(c);
  mf=[abs(c),phasemag(c)];
endfunction
```

priva di tale limitazione. Nella definizione suddetta la tentazione di ricavare la fase con una più familiare **atan(imag(c)/real(c))** non è applicabile, come noto, per il solito problema (del II e III quadrante). L'alternativa corretta sui 4 quadranti è invece **atan(imag(c), real(c))** (che restituisce un angolo in radianti).

e anche da

Polare a Cartesiana

Supposto di avere il numero complesso in forma polare sottoforma di matrice a due colonne (ad es. $z=[100 \ 45]$) anche multidimensionale (ad es. $Ei=[230 \ 90; 230 \ -30; 230 \ -150]$) usiamo la relazione

$z = \rho * (\cos(\theta) + j * \sin(\theta))$ per scrivere una funzione del tipo:

```
//----- polare --> cartesiana -----
function [c] = pc(mf);
    c = mf(:,1).*cos(%pi*mf(:,2)/180) + %i*mf(:,1).*sin(%pi*mf(:,2)/180);
endfunction
```

che ci permette di convertire una matrice reale ... (con modulo e fase su 2 colonne X n righe) nella corrispondente matrice complessa (con 1 colonna X n righe). E usarla con:

```
-->pc(z)
//oppure
-->pc(Ei)
//o abbreviando
-->pc([100 45])
```

Se siamo pigri e vogliamo qualcosa di piu' flessibile dobbiamo complicare la definizione della funzione per far in modo che vengano riconosciute sia la forma del tipo $pc(z)$ sia la forma $pc(m, f)$ molto piu' comoda e quindi una funzione che riconosca sia 1 sia 2 argomenti ecc. !!!

La "**soluzione finale**" al problema puo' essere trovata in uno script per Scilab del prof. Tony Richardson - University of Evansville - [Working with Complex Numbers and Matrices in Scilab](#).

(Attenzione, i nomi delle funzioni, nel documento originale, sono diversi da quelli usati in questo articolo)

(come esercizio x casa; crollate se nei calcoli c'e' qualcosa di non universalmente ... condivisibile! :)

Riporto comunque di seguito uno script leggermente modificato e semplificato.

Ovviamente ... " ... se mi sbaglio ... mi corrigerete ! "

Script 2

```

//----- cut
//          Tasformazioni  Complesse
//-----
      k0=%pi/180; j=%i; // la mia solita pigrizia !
      format('v',6)

//-----
// pc() - Converte da polare a cartesiana.
function [c] = pc(m,f)
  if argn(2) == 1 then
    //Se singolo argomento di ingresso modulo e fase su 2 col. adiacenti
    c = m(:,1:2:~).*cos(m(:,2:2:~)*k0) + j*m(:,1:2:~).*sin(m(:,2:2:~)*k0);
  elseif argn(2) == 2 then
    //Se doppio argomento, modulo e fase in matrici separate
    c = m.*cos(f*k0) + j*m.*sin(f*k0);
  end
endfunction
//
//-----
// cp() - Converte da cartesiana a polare.
function [m,f] = cp(c)
  [nr nc]=size(c);
  if argn(1) == 1 then
    //Se richiesto singolo argomento,
    //modulo e fase ritornano su due colonne adiacenti
    m=zeros(nr,2*nc); //definisco matrice con colonne doppie
    m(:,1:2:~) = abs(c);
    m(:,2:2:~) = phasemag(c);
  elseif argn(1) == 2 then
    //Se richiesto doppio argomento,
    //modulo e fase ritornano su matrici separate
    f = phasemag(c);
    m = abs(c);
  end
endfunction
//----- cut

```

Uso di pc()

Il modo piu' semplice e nello stesso tempo piu' utile

```
-->z=pc(100, -45)
z =
    70.71 - 70.71i
```

ma anche usando un vettore

```
-->e=[5 60; 3 0 ; 6 -45];
-->y=pc(e)
y =
    2.5 + 4.33i
    3.
    4.243 - 4.243i
```

Uso di cp()

Sono diverse le modalita' di utilizzazione della funzione **cp()**, con assegnazione a una o due variabili, usando come ingresso un semplice numero complesso:

```
-->z=3-j*4;
-->y=cp(z)
y =
    5. - 53.13
```

oppure usando un vettore colonna:

```
-->z=[1-j; -10*j; 20; -3-4*j];
-->y=cp(z)
y =
    1.414 - 45.
    10. - 90.
    20. 0.
    5. - 126.9
```

e infine una matrice complessa 2x3

```
-->z=[1-j j 4; 2+j 1-j -3-j*4];
-->y=cp(z)
y =
    1.414 - 45.    1.    90.    4.    0.
    2.236    26.57    1.414 - 45.    5. - 126.9
```

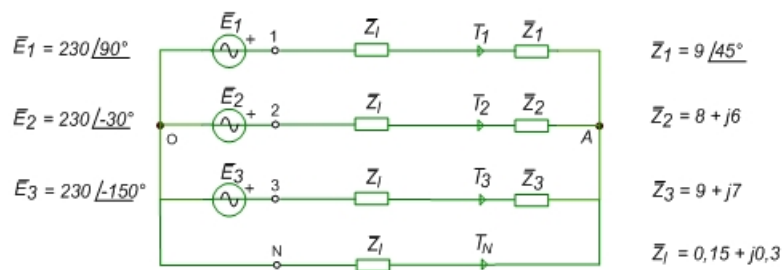
Volendo si possono separare moduli e fasi con

-->[modulo fase]=cp(z)

Esempio

Un esempio applicativo, come al solito, chiarirà le idee.

Supponendo di dover determinare la corrente nel neutro, della seguente rete trifase,



possiamo calcolare $\bar{V}_{AO}$ con Millman :

$$\bar{V}_{AO} = \frac{\frac{\bar{E}_1}{\bar{Z}_1 + \bar{Z}_l} + \frac{\bar{E}_2}{\bar{Z}_2 + \bar{Z}_l} + \frac{\bar{E}_3}{\bar{Z}_3 + \bar{Z}_l}}{\frac{1}{\bar{Z}_1 + \bar{Z}_l} + \frac{1}{\bar{Z}_2 + \bar{Z}_l} + \frac{1}{\bar{Z}_3 + \bar{Z}_l} + \frac{1}{\bar{Z}_l}}$$

e quindi

$$\bar{I}_N = -\frac{\bar{V}_{AO}}{\bar{Z}_l}$$

da Scilab, dopo aver eseguito lo script **Tasformazioni Complesse**, scrivendo :

```
zl=0.15+j*0.3;
Vao=(...
pc(230,90)/(pc(9,45)+zl)+pc(230,-30)/(8+j*6+zl)+pc(230,-150)/(9+j*7+zl))...
/(1/(pc(9,45)+zl)+1/(8+j*6+zl)+1/(9+j*7+zl)+1/zl)
//(zl e' definita esternamente per rendere la scrittura piu' leggibile)
Vao =
    1.111 + 1.739i
In=(-Vao/zl)
```

```

In =
- 6.119 + 0.643i

//e se vogliamo anche le tre correnti di linea
I=[(pc(230,90)-Vao)/(pc(9,45)+zl);
    (pc(230,-30)-Vao)/(8+j*6+zl);
    (pc(230,-150)-Vao)/(9+j*7+zl)]
I =
    17.43 + 17.21i
    8.282 - 20.73i
    - 19.6 + 2.876i
// e in forma polare
Ip=cp(I)
Ip =
    24.49    44.63
    22.32    - 68.22
    19.81    171.7

// infine controlliamo con Kirchhoff :
I(1)+I(2)+I(3)+In
ans =
- 0.000 - 0.000i

```

Per un riferimento storico sul metodo simbolico

[1] C.P.Steinmetz, [*Alternating Current Phenomena*](#),1897.

[2] A.E.Kennelly, [*Tables Of Complex Hyperbolic And Circular Functions*](#),1914.

[3] A.E.Kennelly, [*The Application of Hyperbolic Functions to Electrical Engineering Problems*](#),1914.

Estratto da "<http://www.electroyou.it/mediawiki/index.php?title=UsersPages:Renzodf:articolo2>"